Rule #1	Recommended Implementation	
Avoid spaces	**Do not use spaces within a table or field name.** Many users include spaces in table and field names for easy readability. Instead, use an underscore between words (see **Separate words** rule). Access is one of the only databases to even allow the use of spaces within names, so other databases, development tools, and analysis tools often have difficulty using these fields without custom workarounds added by the programmer or user.	
	Examples of Design	**Examples of Better/Recommended Design**
	Location ID	Location_ID
	Park Code	Project_Code
	Project Code	Park_Code

Rule #2	Recommended Implementation	
Separate word	**Use underscores to separate words within a single table or field name.** Many users rely on mixed case table and field names for easy readability. Rather than relying upon the way the name is typed, a better choice is to separate words with an underscore (_). The name then resembles written text and words within the name are easily identified. Many analysis tools ignore case. When data is imported from a database, the tool converts field names to all upper or all lower case. By using underscores to separate words, the fields maintain readability. However, keep in mind that some databases are case-sensitive. This means that any user must type the names exactly as they have been created to get the expected results.	
	Examples of Design	**Examples of Better/Recommended Design**
	EventID	Event_ID
	StartDate	Start_Date
	EndDate	End_Date

Rule #3	Recommended Implementation	
Consistent case	**Use mixed case text within a table or field name.** All uppercase words are especially straining to the human eye. Mixed case text presents a readable format that is more easily and quickly read.	
	Examples of Design	**Examples of Better/Recommended Design**
	SPECIES_CODE	Species_Code
	SPECIES_COUNT	Species_Count
	behavior_code	Behavior_Code

Rule #4	Recommended Implementation	
Avoid special characters	**Do not use special characters in a table or field name.** Allowable characters include A-Z, a-z, 0-9, and _ (underscore to separate words). Additionally, a name should never begin with a number. Access is one of the only databases to even allow the use of special characters within names, so other databases, development tools, and analysis tools often have difficulty using these fields without custom workarounds added by the programmer or user.	
	Examples of Design	**Examples of Better/Recommended Design**
	Project Park/Region	Project_Park_Region
	Project$	Project_Cost
	ProjLead_Phone#	Proj_Lead_Phone_Num
	Project Complete?	Is_Project_Complete

Rule #5	Recommended Implementation	
Singularize names	**Choose the singular noun form for a field name.**	
	Examples of Design	**Examples of Better/Recommended Design**
	Life_Stages	Life_Stage

Rule #6	Recommended Implementation	
Avoid unknown abbreviations	**Avoid abbreviations unless necessary due to field length.** If an abbreviation is needed, make every attempt to use one that is known within the organization or one that can be easily deciphered. Avoid an abbreviation that is a word itself or has multiple meanings. Some older flat file systems limit table (file) and field names to 8 or 10 characters. Most relational databases have a limit of approximately 30 characters. Since Access and Oracle are the NPS standards, and they do not impose these limits, the use of terse names and extensive abbreviating are not necessary. Take advantage of additional characters to eliminate the ambiguity of table and field names. Keep in mind that in cases where data is exported into a DBF, long field names may need to be re-evaluated.	
	Examples of Design	**Examples of Better/Recommended Design**
	SmpTrnID	Sample_Transect_ID
	Spec_Cd	Species_Code
	VeStCnt	Vertical_Strata_Count

Rule #7	Recommended Implementation	
Limit length	**Limit the length of table and field names to approximately 20 characters.** This limit is set more for practicality than any other reason. Most database servers do have a maximum limit of approximately 30 characters, though. Shorter names can be typed more quickly and are easier to remember. Also, longer names can sometimes extend past the width of set drop down lists, so only the first part of a name is visible. Find an optimal field length where the name is not too tedious, but the name still clearly represents the data stored in it.	
	Examples of Design	**Examples of Better/Recommended Design**
	Water_Quality_Evaluation_Code	H2O_Quality_Eval_Code
	Geomorphic_Disturbance_Description	Geomorphic_Disturb_Desc
	Area_Average_Azimuth_ Magnetic	Area_Avg_Azimuth_Mag

Rule #8	Recommended Implementation	
Specific names	**Choose a name that accurately identifies the data to be stored in a table or field.** Strive to create names that accurately define the data stored within. If a name is too vague, users must rely upon supplemental documentation for definitions. Also, users may enter data that the field was not intended to store.	
	Examples of Design	**Examples of Better/Recommended Design**
	Habitat	Viereck_Class_Code
	Percent_Cover	Tree_Cover_Percent
	Tree_Size	Tree_Height_ft

Rule #9	Recommended Implementation	
Single value	**Choose a field to contain a single value.** Data entry, validation and retrieval are more difficult when a single field contains multiple, independent values.	
	Examples of Design	**Examples of Better/Recommended Design**
	Full_Name	First_Name, Last_Name
	City_State_Zip	City_Name, State_Code, Zip_Code

<table>
<tr><td>Rule #10</td><td colspan="4">Recommended Implementation</td></tr>
<tr><td rowspan="4">Avoid calculations</td><td colspan="4">Choose a field to be independent of all other field values.
Rather than storing a calculated value in the database, a better choice is to store in independent operands and perform calculations dynamically for display within queries, forms, or reports. Stored calculations run the risk of not being updated when one of the individual elements changes.</td></tr>
<tr><td colspan="2">Examples of Design</td><td colspan="2">Examples of Better/Recommended Design</td></tr>
<tr><td colspan="2">Plot_Area_m2</td><td colspan="2">Plot_Width_m, Plot_Length_m</td></tr>
</table>

<table>
<tr><td>Rule #11</td><td colspan="4">Recommended Implementation</td></tr>
<tr><td rowspan="3">Avoid reserved words</td><td colspan="4">Avoid a table or field name that is a word reserved for use by the database server.
Each database server and development environment has a set of reserved words that should not be used as table or field names.
Access, in particular, will allow the creation of a field that is a reserved word. It will often not cause problems until a later time, during the creation and execution of queries, forms, or reports.
Listed are some of the more common reserved words. See the Access online help system for a complete list.</td></tr>
<tr><td colspan="4">Reserved Words</td></tr>
<tr><td>area
avg
by
constraint
count
counter
currency
date
datetime
delete
desc
exists
from
group</td><td>ignore
in
index
insert
key
max
memo
min
money
name
number
option
order
percent</td><td>perimeter
primary
procedure
property
references
report
section
select
set
shape
sum
table
text
time</td><td>type
update
value
values
year
yesno</td></tr>
</table>

Rule #13	Recommended Implementation
Prefix table names	**Prefix each table with the appropriate category abbreviation.** There are three categories of tables. A data table is most common and contains data collected in the field. A lookup table contains a list of valid values that references some other field in the database. A cross-reference (or linking) table is the table created to accurately depict a many-to-many relationship; this table references the two parent tables and contains their primary keys. To help distinguish a table type, prefix the table name with one of the following: • Data Table = tbl_ • Lookup Table = tlu_ • Cross-Reference Table = xref_
	Examples of Recommended Design
	tbl_Locations tbl_Events tlu_Parks tlu_Observers xref_Location_Event xref_Event_Bird_Observation

Rule #14	Recommended Implementation

Prefix objects	**Prefix each object with the appropriate abbreviation.**
	The following list contains some recommended prefixes for other Access objects:
	• Form
	  • Main = frm_
	  • Query (for criteria) = frm_qry_
	  • List (read-only grid) = frm_lst_
	  • Edit (for data entry) = frm_edt_
	  • Sub = frm_sub_
	• Query
	  • Select = qry_
	  • Insert = qry_ins_
	  • Update = qry_upd_
	  • Delete = qry_del_
	• Report
	  • Main = rpt_
	  • Sub = rpt_sub_
	• Macro = mac_
	• Module = mod_